Reactive pipelines for integrated structural bioinformatics resources

Isaure Chauvot de Beauchêne, Sjoerd Jacob de Vries

Reactive pipelines for integrated structural bioinformatics resources

Isaure Chauvot de Beauchêne Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
Sjoerd J. de Vries Parisian Structural Bioinformatics Resource (**RPBS**) platform, INSERM UMR-S973 / Paris Diderot University, France.

Introduction

Integration of structural bioinformatics resources is a major challenge. **FAIR principles** are an excellent direction towards establishing intra-discipline cohesion, and to link up with neighboring scientific disciplines. Eventually, this offers the possibility to move beyond the standard publication model, from manuscripts with imprecise "Results" and "Materials & Methods" sections towards machine-readable scientific resources: interactive, reproducible, searchable, and connectible.

Benefits

- **Reproducibility:** Annotated **"computation trees"** (*see below*) can make computations perfectly reproducible from input
- **Reactivity:** Computation trees can automatically re-compute a structure prediction if any input changes.
- **True interoperability:** by building and combining computation trees.
- **Caching:** if (part of) a computation tree has already been performed, results can be read from data storage.
- **Interactivity:** User can change any data, code, or even the topology, and just evaluate the part of which the dependencies have changed, while the service as a whole is running.
- **Delegated computing:** The computation tree being modular, it can be evaluated in parallel, on many cores: it just wait for its direct dependencies before launching a transformation.

These changes will provide many advantages, but their realisation in **structural bioinformatics** needs to tackle several challenges. This requires adaptation of the FAIR principles toward the realities of structural bioinformatics.

Challenges

- Structural bioinformatics is **fragmented.** Structure prediction tools use a plethora of formats to describe protein models (rotation matrices, normal mode amplitudes, discretized rotamers). When converting to simple atomic coordinates, much information is lost.
- Tools are typically **full-stack protocols**: sampling, scoring and refinement all occur within the same tool, and return few models to be used directly by biologists. This makes tool integration extremely difficult.
- Databases are full of **implicit dependencies**, including time. For example, a PDB code 1XYZ may point to different coordinates over time, changing when the PDB entry gets updated. Then, computations are not reproducible from input parameters with PDB codes.

FAIR principles ...

SCIENTIFIC DATA

Wilkinson et al., 2016. 755 citations

Comment: The FAIR Guiding Principles for scientific data management and stewardship

Findability

(Meta)data are assigned globally unique persistent identifiers
- Data, coordinates: *https://files.rcsb.org/download/1AVX.pdb*
- Data, structure factors: *https://files.rcsb.org/download/1AVX-sf.cif*
- Meta-data: *https://www.rcsb.org/structure/1AVX*
  X-ray resolution, 2D structure, Link to Uniprot ...

Accessibility

(Meta)data are retrievable by their identifier
Metadata are accessible even when data no longer available
Metadata are small; data can be hosted at Dataverse, using torrents...

Interoperability

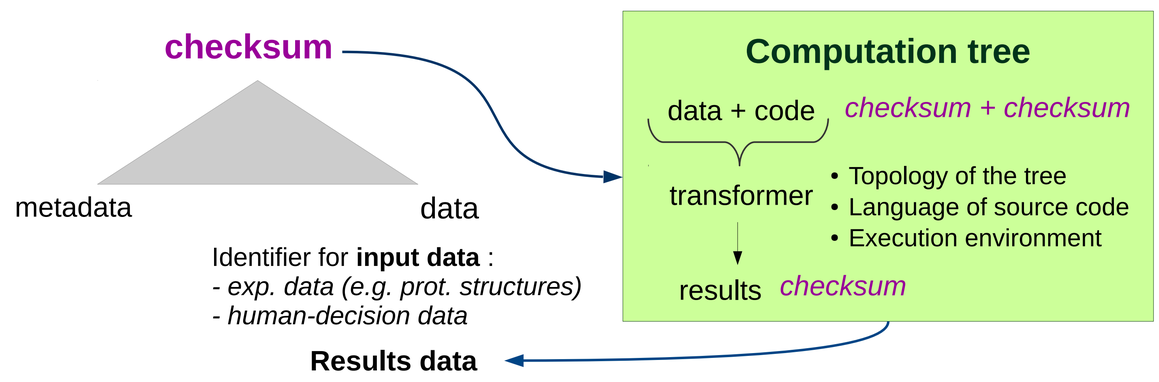
(Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation **(semantic ontology)**

Example: OWL language
DataPropertyAssertion(:id :MyStructure :www.rcsb.org/structure/1PKG)
ObjectPropertyAssertion(:protein :MyStructure :cKIT)
ClassAssertion(:Auto-Phosphorylating-Kinase :cKIT
SubClassOf(:Auto-Phosphorylating-Kinase) ObjectHasSelf(:phosphorylate)

Re-usability

(Meta)data are associated with detailed provenance
Where does it come from? Was it transformed?

... adapted to 3D Bioinformatics



Data should be defined by its **value**, not by a URL. These values should be stored as **checksums***.

Tool interoperability:

A semantic ontology for protein structure is not sufficient to achieve interoperability. **Syntactic ontologies** and their **pairwise conversions** for different protein model formats are necessary.

Labs should focus on **single-purpose tools** that work well and can be integrated by others. Tools should be decomposed into their constituent stages. At each stage, large numbers of models should be kept, to be re-ranked or filtered by downstream tools (e. g. using exp. data).

To be **deterministic and reproducible**, computations should be defined as a **computation tree** of connected data in the form of checksums. Code and metadata are part of the tree.

* A **checksum** is a small-sized datum derived from a block of digital data to verify data integrity. It has significantly different value even for small changes in the digital data.

Perspectives

Technologies developed at the RPBS platform:

A server to map the **checksum** of (meta)data to external URLs where they can be downloaded.

Syntactic ontologies (using a superset of JSON schema) to describe the input and output data formats of structural biology tools.

Implementation of a universal transformer. Its input data uniquely and **deterministically** defines the result of any computation. The transformer is universal: source code is just another kind of input data.

Tracking the dependencies of a computation (incl. code dependencies) into a **computation tree** of data checksums and universal transformations. Any structural biology tool can be decomposed and described in this way.

A server to **map the checksum of a result to its computation tree**. As the inputs are often computation results themselves, this allows a computation to be tracked all the way down to the original experimental data.

Reactive pipelines to re-evaluate computation trees as they change. This allows the automatic re-computation of a structure prediction if any of its inputs change (e. g. because of new experimental data, or if the tool itself is improved).

[1] Lu, Xiang-Jun, Harmen J. Bussemaker, and Wilma K. Olson. "DSSR: an integrated software tool for dissecting the spatial structure of RNA." Nucleic acids research 43.21 (2015): e142-e142.

Example: RNAlib pipeline